

Model *Tracking* Pembicara Dalam Perekaman Video Otomatis Pada Kelas Cendekia

Elga Ridlo Sinatriya^{*1}, Muhammad Idham Ananta Timur², Ika Candradewi³

¹Prodi Elektronika dan Instrumentasi Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM

²Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: ^{*1}elgaridlosinatriya@gmail.com, ²idham@ugm.ac.id, ³ika.candradewi@ugm.ac.id

Abstrak

Kebutuhan kelas cendekia dalam menyimpan informasi yang bersumber oleh pembicara didalam kelas dengan menggunakan konsep ubiquitous computing dimana teknologinya ada akan tetapi tidak terasa disekitarnya. Konsep kelas tersebut membutuhkan beberapa kemampuan diantaranya melakukan tracking terhadap pembicara dan melakukan perekaman. Oleh karena itu dibutuhkan suatu sistem yang mampu mengikuti pembicara secara real – time, menghiraukan pembicara lain, dan merekam kegiatan pembicara terssebut. Waktu yang dibutuhkan oleh sistem dalam melakukan sekali deteksi dan mengirimkan data rata – rata adalah 168.02 ms. Sistem memiliki akurasi sebesar 93.37 % selalu pas tengah hasil rekamannya. Sistem juga mampu mendapatkan akurasi dalam deteksi orang yang benar sebesar 98 %.

Kata kunci—Ubiquitous computing, kelas cendekia, LBP Cascade, Kalman filter

Abstract

The requisite of intelligent classroom's saving the information from speakers inside the class using ubiquitous computing concept. It said the most profound technologies are those that disappear, and they weave themselves into fabric of everyday life until they are indistinguishable from it. It requires a few capability such as tracking the speaker and record it. Therefore it will be require the system that can tracking the speaker in real time, ignore the other speaker, and recording speaker's activity. The system consumes 168.02 ms in one move, like detection using statis camera, send the centroid to microcontroller, second detection using dinamis camera, and record it. The system had an accuracy of 93.37 % to fits the speaker at the middle of frame record. The system is also had an accuracy of 98% to detecting the correct speaker.

Keywords—Ubiquitous Computing, intelligent classroom, LBP Cascade, Kalman filter

1. PENDAHULUAN

Kelas cendekia adalah suatu konsep kelas yang terinspirasi dari keinginan Mark Weiser untuk menciptakan *disappearing computer* yang diambil dari konsep *ubiquitous computing*. Teknologi yang dikatakan matang adalah teknologi yang tidak berasa keberadaannya. Mereka seperti sudah menyatu dalam kehidupan sehari-hari sehingga tidak teradakan [1]. Dari pernyataan Mark Weiser tersebut dikembangkan sebuah kelas yang memanfaatkan konsep *ubiquitous computing* dengan menggunakan komputer sebagai fasilitas dalam berdiskusi dan memiliki kemampuan untuk mengolah informasi.

Kelas cendekia adalah suatu kelas yang menyediakan suatu fasilitas dalam bentuk ruangan untuk berdiskusi. Kegiatan dikelas cerdas [2] ini layaknya kelas diskusi pada umumnya yaitu melakukan suatu tanya-jawab yang memberikan informasi berharga bagi pihak lain. Oleh sebab itu dilakukan penelitian yang mampu mengumpulkan rekaman kegiatan dalam kelas cendekia dan mengurangi campur tangan pengguna terhadap sistem.

Pada dasarnya, orang didalam kelas cendekia tidak perlu banyak melakukan campur tangan pada sistem yang digunakan untuk merekam semua kegiatan mereka, dan menggerakkan kamera kearah yang berbicara. Pada kelas cendekia ini orang yang berdiskusi hanya duduk dan melakukan tugas mereka yaitu bertanya – jawab (berdiskusi). Kamera akan bergerak secara otomatis untuk mengikuti dan merekam kegiatan pembicara [3].

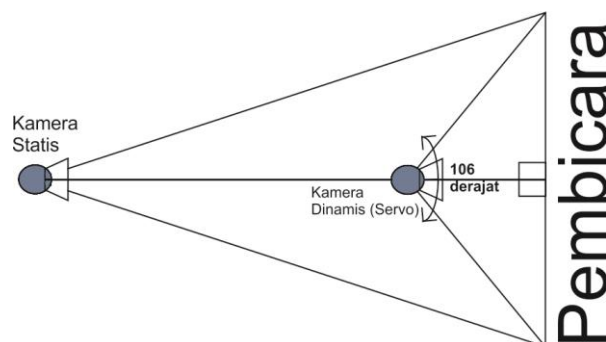
Kelas cendekia memerlukan suatu sistem yang dapat merekam kegiatan dalam kelas cendekia secara otomatis. Pada kelas cendekia, pembicara dalam keadaan duduk yang membuat kamera hanya dapat melihat bagian *upper-body* pembicara, maka deteksi yang bisa dilakukan untuk mendeteksi posisi pembicara adalah dengan cara mendeteksi wajah pembicara [4]. Wajah merupakan bagian tubuh yang mudah dikenali.

Terkadang saat dilakukannya perekaman, ada kecenderungan pembicara tidak bisa diam saat perekaman, atau bisa dibilang bergeser dari posisi semula. Hasil rekaman video yang bagus adalah mendapatkan seluruh tubuh bagian atas pembicara, karena saat pembicara bergeser akan membuat kamera tidak menangkap dengan hasil yang bagus. Oleh karena itu dibutuhkan sistem yang mampu menggerakkan kamera untuk melakukan *tracking* gerakan [5] pembicara dan merekamnya, serta mampu berfokus kepada pembicara yang sudah ditetapkan.

2. METODE PENELITIAN

Sistem deteksi yang diimplementasikan pada kelas cendekia yaitu melakukan deteksi wajah pada pembicara yang kemudian dilakukan penguncian pada satu wajah yang akan diikuti (*tracking*). Sistem deteksi wajah menggunakan LBP (*Local Binary Pattern*) *cascade* yang terdapat pada pustaka openCV 2.4.13.2 dengan bahasa C++. Kemudian untuk melakukan *tracking* terhadap pembicara menggunakan *Kalman filter*. Ada dua buah kamera yang digunakan pada sistem *tracking* [6] dan perekaman video otomatis ini, yaitu pertama kamera yang digunakan untuk mendeteksi pembicara (kamera statis) dan kamera yang digunakan untuk mengikuti pembicara serta merekam (kamera yang berada diatas servo). Posisi kamera statik dan kamera servo dapat dilihat pada

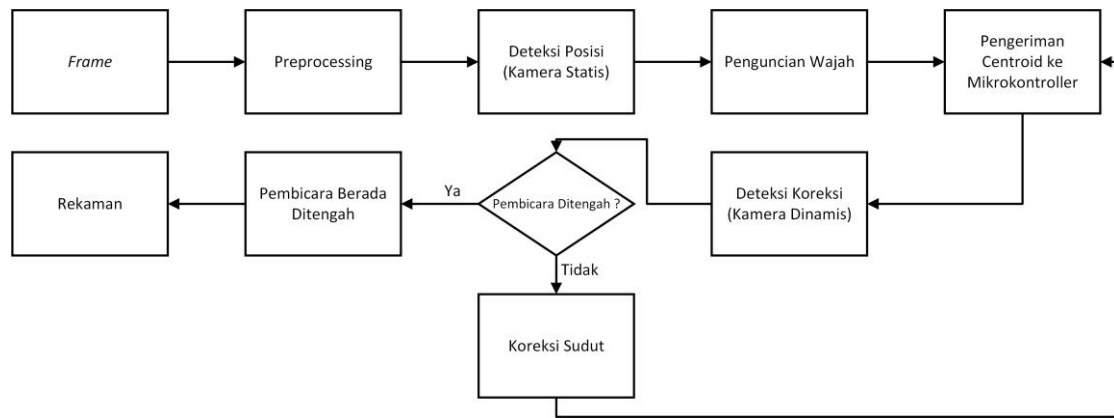
Gambar 1.



Gambar 1 Posisi letak kamera statik dan kamera servo

2.1 Deskripsi Sistem

Diagram blok pada Gambar 2 merupakan penjelasan cara kerja sistem deteksi dan *tracking* dalam kelas cendekia. Berawal dari masukan informasi berupa citra yang didapatkan dari kamera statis sampai menggerakkan servo dari deteksi titik tengah pembicara.



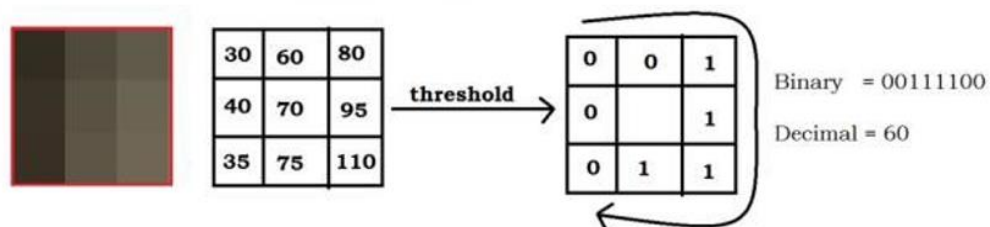
Gambar 2 Diagram Blok Sistem

Kegiatan pertama adalah mengambil citra dengan menggunakan kamera, kemudian citra tersebut dimasukkan kedalam *pre-processing* yang terdiri dari *grayscale* dan *histogram*. *Grayscale* digunakan untuk membuat citra menjadi berwarna hitam dan putih (keabu-abuan), kemudian citra yang sudah diubah menjadi *grayscale* dibawa ke *histogram*, disini citra *grayscale* dilakukan pengaturan untuk meningkatkan kecerahan dalam pengolahan citra. Pada pendeteksian wajah, *histogram* akan mampu untuk membuat perbedaan warna *background* yang gelap dan warna wajah yang biasanya lebih cerah.

Citra wajah yang sudah ditingkatkan kecerahannya dibandingkan dengan *data set LBP cascade* yang terdapat didalam pustaka openCV. Apabila saat dibandingkan sesuai maka citra tersebut terdapat wajah didalamnya. Wajah yang terdeteksi tersebut akan diberi titik tengah yang disebut *centroid* wajah. Titik tengah tersebut yang digunakan untuk mengunci wajah pembicara yang diikuti dan menghiraukan wajah yang lain. Metode *kalman filter* digunakan untuk melakukan *tracking* atau prediksi ketika tidak terdapat wajah pada ROI penguncian, maka *kalman* akan mencari titik tengah tadi dengan melakukan perbandingan nilai *centroid* sebelumnya dan *centroid* sekarang. Nilai *centroid* ini digunakan untuk menggerakkan servo yang melakukan perekaman dan *tracking* kepada pembicara.

2. 2 Local Binary Pattern

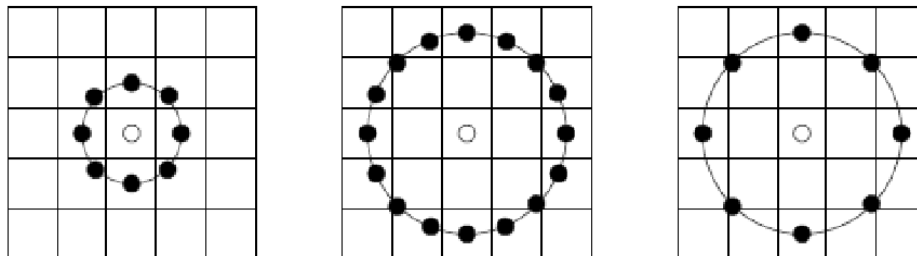
Metode ini menetapkan sebuah label untuk setiap piksel dari citra dengan cara memberikan *thresholding* atau batasan yaitu 3x3 piksel dengan membandingkan nilai piksel tengah dengan piksel tetangga luarnya yang berhimpitan langsung. Hasil yang didapatkan akan menjadi bentuk biner yang kemudian dikonversi menjadi bentuk desimal.



Gambar 3 Operasi Local Binary Pattern [7]

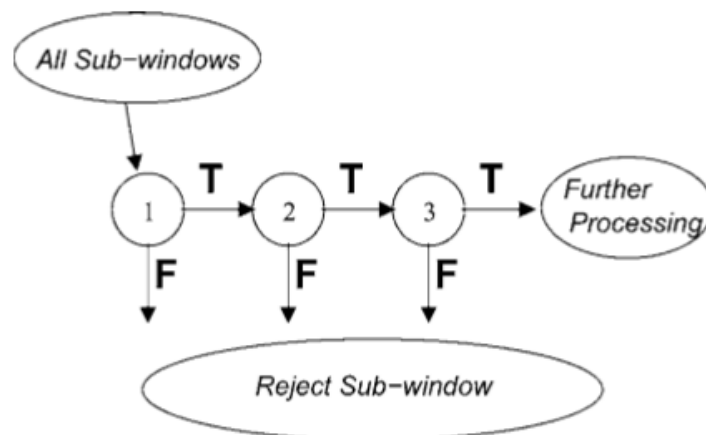
Gambar 4 merupakan notasi (P,R), P berartikan *sampling point* atau titik sampling pada lingkaran dengan radius R. Pelebaran dari lingkaran tersebut dinamakan *uniform pattern*. LBP disebut sebagai *uniform pattern* ketika nilai biner terdapat didalamnya paling banyak 2 bitwise transisi mulai dari 0 ke 1 atau sebaliknya. Semisal binernya berpola 00000000 terdapat 0

transisi, 00001111 terdapat 2 transisi, 01100000 terdapat 2 transisi. Dua transisi tersebut yang dinamakan dengan *uniform*. Pada komputasi pelabelan dari LBP, untuk setiap *uniform pattern* diberikan label terpisah atau tersendiri dan untuk semua *non-uniform pattern* diberikan label yang sama. Ketika menggunakan $(8,R)$ neighborhood terdapat total 256 pola, dengan 58 pola *uniform* dan terdapat 59 label yang berbeda [7].



Gambar 4 Notasi (P,R) [7]

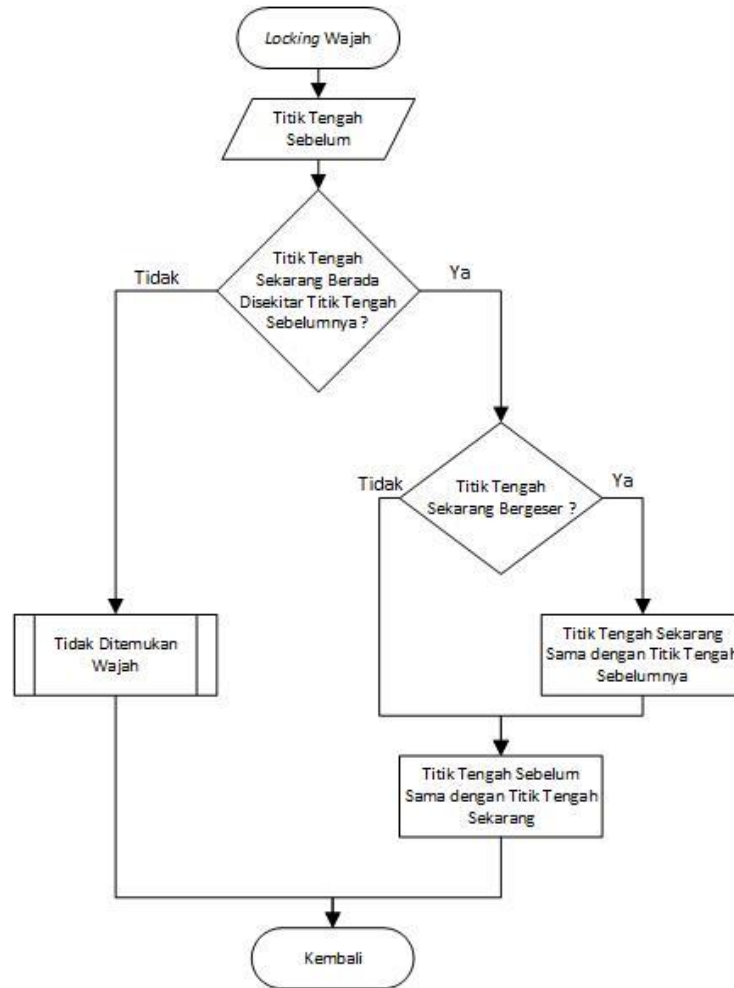
Cascade classifier digunakan untuk meningkatkan kemampuan deteksi dan mengurangi jumlah komputasi. *Weak classifier* digunakan untuk menolak *sub-window* dari citra yang dibagi tadi sebelum klasifikasi semakin kompleks yang biasanya mendapatkan tingkat kesalahan yang lebih rendah. Setiap tahapan didalam *cascade* sudah disusun dan dilatih oleh *Adaboost*. Hasil positif dari *strong classifier* diawal akan masuk ketahap kedua, kemudian akan masuk ketahap tiga dan seterusnya. Sedangkan untuk hasil negatif akan langsung ditolak pada tahapan yang diketahui bernilai negatif [8].



Gambar 5 Skematik dari Deteksi Cascade [8].

2.3 Algoritma Penguncian Wajah

Penguncian pada wajah digunakan untuk menghiraukan wajah-wajah lain yang terdeteksi didalam citra. Wajah yang dikunci inilah yang akan diikuti oleh servo dan direkam oleh kamera diatas servo. Gambar 5 merupakan algoritma penguncian wajah dengan cara membandingkan nilai titik tengah pada wajah, apakah masih berada pada disekitar titik tengah ROI. Apabila tidak berada disekitar titik tengah ROI, maka dianggap wajah tidak ditemukan.



Gambar 6 Alur Algoritma Penguncian Wajah

2.4 Metode Kalman Filter

Metode *Kalman filter* digunakan untuk melakukan prediksi [9] dimana algoritma penguncian tidak mampu mengikuti wajah pembicara. Model *kalman filter* dapat memberikan estimasi *state* dengan melihat dua tahap utama, yaitu tahap update waktu dan tahap koreksi. Tahap pertama filter akan memprediksi titik sekarang (*state current*) dengan menggunakan observasi dari *state* sebelumnya. Tahap kedua merupakan tahap koreksi, filter akan menyaring prediksi dan mengoreksi estimasi dari *state* dengan cara mengambil kedalam pengukuran sekarang (*actual measurement*). Kedua tahap tersebut dapat dijadikan menjadi persamaan, dimana persamaan (1) sebagai tahap prediksi dan persamaan kedua (2) merupakan tahap [10].

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (1)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (2)$$

Dimana variabel \hat{x}_k^- adalah estimasi prior pada *frame* ke k , \hat{x}_k adalah estimasi posterior pada *frame* ke k , u_{k-1} adalah ukuran /lokasi pada *frame* ke k . K_k adalah Kalman gain pada *frame* ke k , sedangkan A, B, H adalah matriks dengan dimensi yang berbeda. Dari persamaan (1) dan (2), dilihat keluaran dari tahap prediksi akan digunakan sebagai masukan untuk tahap koreksi.

Dimana

$$x = x, y, v_x, v_y, w, h \quad (3)$$

$$z = z_x, z_y, z_w, z_h \quad (4)$$

$$A = \begin{bmatrix} 1 & 0 & dT & 0 & 0 & 0 \\ 0 & 1 & 0 & dT & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

x merupakan masukan berupa koordinat x dan y yang memperlihatkan posisi tengah dari x -axis dan posisi tengah dari y -axis. v_x, v_y adalah *velocity* dari x -axis dan y -axis. Kemudian terdapat variabel w dan h yang diketahui sebagai tinggi dan lebarnya wajah pembicara. Variabel matriks A menjelaskan transisi dari matriks. Variabel matriks H matriks pengukuran, dan T interval waktu antara dua *frame*. w_k, v_k adalah *Gaussian noises* dengan kovarian *error*.

Tahap – tahap pada Kalman *filter* yang dilakukan

- Waktu *update* dari bagian yang di estimasikan

$$x_{k|k-1} = Ax_{k-1|k-1} \quad (8)$$

- Predicted measurement

$$z_{k|k-1} = Hx_{k|k-1} \quad (9)$$

- Time update of the state error covariance

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q_k \quad (10)$$

- Kalman gain

$$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \quad (11)$$

- *Update pengukuran dari state error covariance*

$$P_{k|k} = (I - KH)P_{k|k-1} \quad (12)$$

- *Update pengukuran dari state estimate*

$$x_{k|k} = x_{k|k-1} + K_k(z_k - z_{k|k-1}) \quad (13)$$

3. HASIL DAN PEMBAHASAN

Dari hasil implementasi dilakukan pengujian yang dilakukan untuk mendapatkan kehandalan sistem untuk mengunci wajah pembicara, prediksi gerakan ketika wajah tidak berada dalam ROI, dan mendapatkan rekaman yang pembicaranya berada ditengah.

Pada Gambar 7 merupakan hasil dengan menggunakan algoritma penguncian, terlihat bahwa penguncian tersebut sangat efektif. Algoritma penguncian mampu mengunci terhadap satu wajah yang akan diikuti. walaupun penguncian tersebut dilakukan dengan latar belakang yang berbeda beda.

Kelemahan dari algoritma penguncian tersebut adalah ketika wajah pembicara yang dikunci terhalangi wajah pembicara yang baru, maka ada kemungkinan wajah baru tersebut yang akan diikuti. Hal tersebut terjadi karena setiap wajah tidak memiliki identitas khusus, identitas pada wajah hanya diambil dari letak koordinat (x,y) pada *frame*.



Gambar 7 Hasil penguncian terhadap pembicara

Pada Tabel 1 menunjukkan hasil akurasi penguncian dengan menggunakan algoritma penguncian rata – rata akurasi yang didapatkan terbilang sangat tinggi yaitu 98.00 %.

Tabel 1 Akurasi penguncian

| Percobaan ke- | Jumlah <i>Frame</i> | Akurasi |
|---------------|---------------------|---------|
| 1 | 77 | 100 % |
| 2 | 362 | 96.96 % |
| 3 | 242 | 100 % |
| 4 | 494 | 97.77 % |
| 5 | 558 | 98.75 % |
| 6 | 203 | 97.04 % |
| Rata-rata | 1936 | 98.00 % |

Percobaan kedua merupakan pengujian saat wajah tidak terdeteksi. *Kalman filter* akan mulai bekerja ketika wajah tidak terdeteksi pada *frame*. Wajah tidak terdeteksi disekitar titik tengah ROI hal tersebut disebabkan oleh beberapa faktor salah satunya adalah pembicara bergerak terlalu cepat sehingga algoritma penguncian tidak dapat mengikuti.

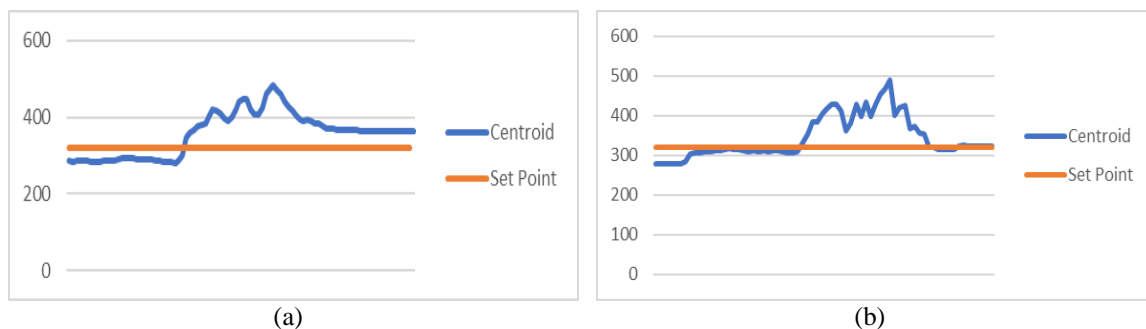
Pada Gambar 8 dapat dilihat bahwa ketika wajah pembicara yang dikunci bergerak terlalu cepat, maka dengan *Kalman filter* mulai bekerja dan mencari dengan menggunakan masukan nilai sebelumnya yang sudah dihitung dengan nilai sekarang. Kemudian nilai *state post* yang didapatkan akan digunakan sebagai masukan yang baru untuk *centroid*. Kemudian diakhir gambar terlihat bahwa *centroid* mampu menemukan kembali wajah pembicara tersebut.

Kekurangan dari sistem ini adalah *Kalman filter* hanya dapat melakukan pencarian secara linier. *Kalman filter* tidak dapat menemukan wajah pembicara apabila pembicara merubah pergerakannya secara cepat dan mendadak.



Gambar 8 Hasil prediksi lokasi pembicara dengan *Kalman filter*

Pengujian ketiga adalah melihat kesesuaian letak rekaman pembicara sudah berada ditengah garis $x = 320$ vertikal pada *frame*. Pengujian tersebut dilakukan menggunakan kamera dinamis, karena kamera dinamis yang digunakan untuk merekam. Pengujian ini juga terdapat dua percobaan yaitu ketika sistem menggunakan kontrol proporsional dan tanpa kontrol proporsional.



Gambar 9 Grafik (a) Tanpa kontrol proporsional dan (b) Dengan kontrol proporsional

Pada Gambar 9 dapat dilihat grafik yang tidak menggunakan kontrol proposional tidak diberikan *feedback* kesalahan deteksi, maka didapatkan hasil yang kurang presisi. Sedangkan untuk grafik dengan kontrol proporsional terlihat *centroid* hasil deteksi menggunakan kamera dinamis berhimpitan pada nilai 320.

Tabel 3 Tidak menggunakan kontrol proporsional vs menggunakan kontrol proporsional

| No | Tidak menggunakan Kp | Menggunakan Kp |
|-----------|----------------------|----------------|
| 1 | 86.31 % | 98.75 % |
| 2 | 79.78 % | 98.53 % |
| 3 | 84.92 % | 98.86 % |
| 4 | 86.84 % | 97.17 % |
| 5 | 87.89 % | 98.53 % |
| Rata-rata | 85.15 % | 93.37 % |

Pada Tabel 3 dapat dilihat bahwa saat sistem diberikan kontrol sistem akan lebih akurat daripada tidak diberi kontrol. Hal itu dikarenakan saat diberikan kontrol berarti sistem diberikan *feedback*. Umpan balik didapatkan dari deteksi error oleh kamera dinamis.

Adapun kemampuan sistem untuk mengejar posisi pembicara. Saat pengejaran sistem akan mendeteksi melalui kamera statis, kemudian nilai tersebut langsung dikirim kepada mikrokontroller. Kemudian kamera dinamis akan mencocokkan kesesuaian letak pembicara yang harus berada pada garis tengah vertikal pada *frame*. Pada tabel 4 hasil yang didapatkan untuk sekali melakukan perintah diatas adalah 168.02 ms.

Tabel 4 Total Waktu Keseluruhan dan Rata – Rata Sekali Deteksi

| Pengujian | Total Waktu (ms) | Rata - Rata Sekali Deteksi (ms) |
|------------|------------------|---------------------------------|
| 1 | 5056 | 168.53 |
| 2 | 5344 | 167 |
| 3 | 5698 | 167.59 |
| 4 | 5036 | 167.87 |
| 5 | 5243 | 169.13 |
| Rata rata: | 5275.4 | 168.02 |

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan seperti waktu komputasi pada setiap sistem mendeteksi wajah, mengunci, mengirim masukan kepada mikrokontroller, dan pendeteksian kedua. Rata – rata diperoleh waktu sebesar 168.02 ms.

Penggunaan algoritma penguncian terhadap sistem sangat efektif untuk menghiraukan banyaknya wajah yang terdapat pada *frame*, ketika sudah mendapatkan wajah yang dikunci sistem mampu mencapai akurasi penguncian sebesar 98.00%. Kontrol proporsional yang dimasukkan kedalam sistem dapat mengatasi kendala dimana wajah pembicara tidak berada pada garis tengah vertikal *frame* dengan data akurasi tanpa kontrol proporsional adalah 85.15% dan dengan kontrol proporsional adalah 93.37 %, karena saat menggunakan kontrol proporsional sistem mendapatkan masukan berupa *error*.

5. SARAN

Pada penelitian ini terdapat beberapa hal yang perlu disempurnakan. Saran-saran yang dapat digunakan untuk menyempurnakan penelitian selanjutnya seperti perlunya *face recognition* untuk memberikan identitas kepada setiap wajah yang dideteksi dan untuk mengatasi kendala dimana terdapat wajah yang menghalangi wajah pembicara agar sistem penguncian tidak mengunci wajah lain yang masuk. Kemudian diperlukannya algoritma yang

mampu melakukan kalibrasi otomatis untuk membuat sistem tidak perlu dilakukan kalibrasi secara manual. Kemudian perlunya mengadopsi konsep IOT sehingga koordinasi antar kamera dapat dikembangkan dengan skala lebih besar.

DAFTAR PUSTAKA

- [1] E. Oran, "Integrating Mobile and Ubiquitous Computing in a Smart Classroom to Increase Learning Effectiveness," *Int. Conf. Educ. e-Learning Innov.*, pp. 1–5, 2012 [Online]. Available: <http://ieeexplore.ieee.org/document/6360684/> [Accessed: 27-Aug-2017]
- [2] J. Park, K. An, D. Kim, and J. Choi, "Multiple human tracking using multiple kinects for an attendance check system of a smart class," *2013 10th Int. Conf. Ubiquitous Robot. Ambient Intell. URAI 2013*, vol. 25, no. 10, p. 130, 2013 [Online]. Available: <http://ieeexplore.ieee.org/document/6677494/> [Accessed: 29-Aug-2017]
- [3] T. Tsuda, M. Okuda, K. Mutou, and Y. Nishida, "Automatic tracking camera system utilizing the position of faces in the shot image," *IEEE Conf. Publ.*, 2006 [Online]. Available: <http://ieeexplore.ieee.org/document/4150108/> [Accessed: 29-Aug-2017]
- [4] D. E. Pratiwi and A. Harjoko, "Implementasi Pengenalan Wajah Menggunakan PCA (Principal Component Analysis)," *Ijeis*, vol. 3, no. 2, pp. 175–184, 2013 [Online]. Available: <https://jurnal.ugm.ac.id/ijeis/article/view/3892> [Accessed: 29-Aug-2017]
- [5] P. Chong, "A Novel Pedestrian Detection and Tracking with Boosted HOG Classifiers and Kalman Filter," *IEEE Student Conf. Res. Dev.*, pp. 1–5, 2016 [Online]. Available: <http://ieeexplore.ieee.org/document/7810052/> [Accessed: 29-Aug-2017]
- [6] C. Miller, B. Allik, M. Ilg, and R. Zurakowski, "Kalman filter-based tracking of multiple similar objects from a moving camera platform," *Proc. IEEE Conf. Decis. Control*, pp. 5679–5684, 2012 [Online]. Available: <http://ieeexplore.ieee.org/document/6425956/> [Accessed: 29-Aug-2017]
- [7] K. Kadir, M. K. Kamaruddin, H. Nasir, S. I. Safie, and Z. A. K. Bakti, "A comparative study between LBP and Haar-like features for Face Detection using OpenCV," *2014 4th Int. Conf. Eng. Technol. Technopreneuship, ICE2T 2014*, vol. 2014–August, pp. 335–339, 2015 [Online]. Available: <http://ieeexplore.ieee.org/document/7006273/> [Accessed: 29-Aug-2017]
- [8] J. Chang-yeon, "Face Detection using LBP features," *Stanford.Edu*, pp. 1–4, 2008 [Online]. Available: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Face+Detection+using+LBP+features#0> [Accessed: 10-Jul-2017]
- [9] S. Sarwar, N. I. Rao, and M. F. Khan, "Real-time object tracking using Powell's direct set method for object localization and kalman filter for occlusion handling," *2012 Int. Conf. Digit. Image Comput. Tech. Appl. DICTA 2012*, pp. 1–7, 2012 [Online]. Available: <http://ieeexplore.ieee.org/document/6411705/> [Accessed: 29-Aug-2017]
- [10] I. Abbas and P. Ali, "Small Target Detection and Tracking Based on the Background Elimination and Kalman Filter," *2015 Int. Symp. Artif. Intell. Signal Process.*, pp. 328–333, 2015 [Online]. Available: <http://ieeexplore.ieee.org/document/7123509/> [Accessed: 29-Aug-2017]